Project 1, Milestone 2 – Multicore scaling studies

Use OpenMP to parallelize the advection code in Milestone 1. You are strongly encuraged to thoroughly debug your Milestone 1 program before proceeding. A few requirements of the code:

- Use the `omp_get_wtime()` function to do performance measurements.
- Use the `default(none)` clause for parallel regions, which then requires you to explicitly declare all variables as either private or shared.
- Parallelize your initial condition routine as well as the main loop.
- Parallelize by having each core carry out a roughly equal subset of the iteration space. You can do this in different ways – explicit loop bound calculations in a paarallel regions, nested omp for clauses, and dividing up the iteration space in slabs, squares, etc. The choice is yours, but you should try various options to make sure your approach doesn't negatively impact performance relative to a different choice.
- Experiment with the different schedule clauses and associated block sizes and do your final timing studies with the best combination.

In your report, include the following:

- details of your processor and compiler choice for any data or plots shown.
- a demonstration that you get the same answer with the parallel and serial versions. Is bitwise reproducibility expected?
- your (best) grind rate (timesteps/s) using the parameters from Milestone 1 but on a problem with N=10,000. This will be the standard value that we compare with each other in class.
- A plot of a strong scaling analysis using the following parameters:
  - N = 3200
  - NT = 400
  - L = 1.0
  - T = 1.0e3
  - u = 5.0e-7
  - v = 2.85e-7
- A plot of a second strong scaling study, using the same parameters as above but with N=200.
- A plot of a weak scaling analysis using the following parameters:
  - N = 800 (for 1 thread – increase as you add threads)
  - NT = 400
  - L = 1.0
  - T = 1.0e3
  - u = 5.0e-7
  - v = 2.85e-7